DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9621

Activity Based Costing Techniques for Workload Characterization

by

Guido Dedene

Herlinde Leemans

Katholieke Universiteit Leuven

Naamsestraat 69, B-3000 Leuven

ONDERZOEKSRAPPORT NR 9621

Activity Based Costing Techniques for Workload Characterization

by

Guido Dedene
Herlinde Leemans

D/1996/2376/21

Activity Based Costing Techniques for Workload Characterization

Guido Dedene
Katholieke Universiteit Leuven
Faculty of Economics and Applied Economics
Naamsestraat, 69
B-3000 Leuven
Belgium

Fax : + 32 16 32 67 32
E-mail : Guido.Dedene@econ.kuleuven.ac.be

Herlinde Leemans
Katholieke Universiteit Leuven
Faculty of Economics and Applied Economics
Naamsestraat, 69
B-3000 Leuven
Belgium

Fax : + 32 16 32 67 32
E-mail : Herlinde.Leemans@econ.kuleuven.ac.be

Abstract

This paper addresses the problem of non-captured service demands in workload monitoring data. Capture ratios are the coefficients that correct the workload service demands so that they fit the global system monitoring data. This paper proposes new techniques for the determination of capture ratios by means of Activity Based Costing techniques. The techniques are illustrated by means of a case study, which also illustrates the non-trivial nature of capture ratios in practical performance analysis.

Introduction

The validation of computer performance models and the calibration of the model parameters is one of the hard tasks in building practical performance models for operating systems. In particular bottleneck analysis cannot be done correctly without accurate visit ratio numbers (or the service demands) for all workloads [Lazowska 84].

Practical performance management uses two input sources for monitoring data for the analysis of the performance of a particular system. On one hand, global system monitors measure the occupation of the resources in the configuration, at least as they are exploited by the operating system. A global system monitor results in system-wide monitoring data for each resource. On the other hand, workload monitors collect performance data for the individual workloads, such as an online time-sharing environment, a database system, a Batch workload, a transaction workload and so on. Workload monitors, which are mostly pretty specific and tailored to the performance criteria of a particular workload type, produce (amongst other information) performance data on the resource utilisation, for that particular workload type.

Although it seems natural that a performance analyst can start from these measurements to determine the workload service demands for each of the resources, additional corrections on the monitoring data may be needed. Due to the fact that a workload monitor can only monitor the resources from the own viewpoint of the workload, some activity may not be captured. In fact, the activity in the operating system which takes place in supervisor mode may be difficult to assign directly to individual workload types during the monitoring process itself. As a result, the sum of all the workload measurements does not match the global performance data from the system monitors in many cases. This underestimation of the service demands on resources by workload types in the workload monitoring measurements creates difficulties in performance model creation and calibration. More precisely, if the service demand for some device k by workload j, as measured by a workload monitor for class j, is denoted as $D_{k,j}$, and if D_k denotes the total service demand for the device k as measured by some global system monitor, then the following can be observed

$$\sum_{j=1}^{C} D_{k,j} \leq D_k$$

where C is the total number of workloads. The service demands $D_{k,j}$ as measured by the workload monitors have to be corrected by capture ratios $CR_{k,j}$ for each device k and workload j to give the corrected service demands $D^*_{k,j}$. Concretely, the capture ratios $CR_{k,j}$ are defined by

$$CR_{k,j} = \frac{D_{k,j}}{D^*_{k,j}} \leq 1$$

such that

$$\sum_{j=1}^{C} \frac{D_{k,j}}{CR_{k,j}} = D_k$$

which simply means that the sum of the service demands by workload, corrected by the capture ratios, should fit the total service demand over all workloads.

Although this fact is most clearly observed in multitasking operating systems, the first research on capture ratios started in the early eighties [Bourret 80]. Most performance monitoring handbooks present only very global approaches for the correction that is required, such as the use of a global overall correction factor. Such a global factor ignores actually the proper causes of capture ratios, namely the occurance of system events related to paging, swapping and I/O, for example, whose service demands cannot directly be attributed during the measurement process itself. Interactive workloads typically suffer more from this type of effects compared to batch workloads. A major reason for this is the fact that interactive workloads have a higher memory competition, resulting in typical higher Paging and Swapping activity, which causes more system interventions. The case study later in this paper will confirm this. In particular, with an increasing use of Client/Server processing, where a lot of the intermediate processing and coordination happens in supervisor mode on both Client and Server systems, incorrect workload monitoring is more likely to occur. This means that without the proper capture ratios most Client/Servers models will suffer from underestimated service demands by workload type.

Consequently, more refined techniques are required to obtain more accurate service demands for workload characterization in computer performance modeling and capacity planning. This paper will formulate and discuss various techniques for the determination of the capture ratios. First a straightforward overhead-based technique is presented. This technique is next extended by using a regression-based technique. Some difficulties related to the usage of multiple regression analysis are discussed. The main contribution, apart from the comparative analysis of the techniques, is the proposal of a new method for the calculation of the capture ratios, based on ideas inherited from Activity Based Costing techniques in accounting [Hongren, Foster and Datar 94, Cooper & Kaplan 92]. The various methods of this paper are illustrated by a case study on CPU measurements in a mainframe server system. This case study is not restricted to this situation, as capture ratios have been observed in a variety of operating systems.

The correctness of the proposed capture ratios, and in particular the methods by means of which they are calculated, can be validated as follows. First of all, measurements on systems with one single workload represent extreme simple measurement situations. On the other hand, the mix of different workloads will require more refined techniques, as in a single workload system all techniques are global, de facto. Secondly, the capture ratios are required to be reproducible and stable over time for one particular system with a fixed mix of workloads. The capture ratios are also related to the degree of system activity required by the workload. System parameters are typically used to tune these interventions. Hence, capture ratio effects can be correlated to tuning parameter settings of the operating system (such as the lengths of priority feedback performance periods for interactive workloads).

Although in literature capture ratios have been studied primarily for CPU-measurements it should be noted that capture ratios also occur in memory and disk measurements. The correct determination of capture ratios can contribute to achieving more correct computer performance models. Correctness means in this situation conformity with the observed performance measurements. Performance modeling without incorporating capture ratios leads to performance models that are difficult to validate.

Global Capture Ratio Determination

A straightforward approach is the correction of the uncaptured service demands by means of the relative total occupation by resource by class. More precisely, if

$D_{k,j}$: the service demand for workload j on device k as measured by a workload monitor for the workload type of class j.

D_k : the total service demand on device k as measured by a system monitor

$D_{k,tot}$: the total "captured" service demand on device $k = \sum_{j=1}^{C} D_{k,j}$

$D_{k,ovh}$: the non-captured service demand on device $k = D_k - D_{k,tot}$

then every workload j can be apportioned a fraction of the non-captured service-demand. This means that the service demand for every workload j is corrected by an additional demand

$$Dovh_{k,j} = \left(\frac{D_{k,j}}{D_{k,tot}} \right) \times D_{k,ovh}$$

which results in a total corrected service demand for device k by workload j :

$$D^*_{k,j} = D_{k,j} + Dovh_{k,j} = D_{k,j} \times \left(1 + \frac{D_{k,ovh}}{D_{k,tot}} \right)$$

Hence the capture ratio for workload j is given by

$$CR_{k,j} = \frac{D_{k,j}}{D^*_{k,j}}$$

$$= \frac{D_{k,j}}{D_{k,j} + \frac{D_{k,ovh}}{D_{k,tot}}} = \frac{D_{k,tot}}{D_{k,tot} + D_{k,ovh}} = \frac{D_{k,tot}}{D_k}$$

This last expression is independent of j, which just confirms that under this approach all workload types have the same capture ratio. This approach was promoted in a number of publications [Wicks 89, Irwin 83]. It is even used implicitly in a number of monitors (to calculate the resource occupations by workload) and performance modeling tools. Nevertheless the above analysis shows that this technique is too simple to be reliable in practice. It ignores, for example, completely the individual nature of different workload types as it does not take into account the kind of system supervisor calls that a workload type is using.

Capture Ratio Determination by Multiple Regression

An alternative approach for the determination of the capture ratios consists of formulating the capture ratio determination problem as a multiple regression. The starting point is the basic relation for capture ratios :

$$\sum_{j=1}^{C} \frac{D_{k,j}}{CR_{k,j}} = D_k$$

which can be reformulated as an equation with C unknown variables

$$\sum_{j=1}^{C} X_{k,j} \times D_{k,j} = D_k$$

if $X_{k,j} = 1/CR_{k,j}$. Of course the unknown variables $X_{k,j}$ must satisfy the boundary condition

$$X_{k,j} \geq 1$$

since capture ratios can be at most 100%. To obtain a solution for the unknown variables $X_{k,j}$ several measurements must be collected to set up a system with at least C equations. In concrete the following problem can be formulated :

$$\sum_{j=1}^{C} X_{k,j} \times D_{k,j,t} = D_{k,t}$$

$$X_{k,j} \geq 1$$

where $D_{k,j,t}$ is the service demand on device k by workload j as measured by a workload monitor for j during time interval t, and $D_{k,t}$ is the total service demand on device k as measured by a system monitor during interval t.

Although this approach seems very natural, its application suffers from a number of difficulties. First of all, standard statistical packages have no procedures for multiple regression subject to boundary conditions. The best alternative is to try a model fitting procedure, starting from seed points that are within the range of acceptable values. In SAS the template for such a procedure is given by:

```
PROC NLIN    DATA = ....   METHOD = GRADIENT;
PARMS        X1 = 1.4
             X2 = 1.4
             ...
             XC = 1.4;
BOUNDS       X1 > 1
             X2 > 1
             ...
             XC > 1;
MODEL DK = X1*DK1 + ... + XC*DKC.
```

The output delivers the required coefficients Xj (j = 1 ... C). Classical instruments to evaluate the intrinsic quality of the regression model (such as an R^2-coefficient) are not meaningful here, due to the presence of boundary conditions.

Another difficulty that is well-known in multiple regression models is the presence of coefficients that have a high degree of correlation (or anti-correlation). One approach to solve this difficulty is to group the correlating workloads into one larger workload type. Another technique is the introduction of a "noise" variable F_k , which has by construction a trivial independence of the other coefficients in the multiple regression model. The capture ratio determination with a noise factor is done by solving the problem:

$$\sum_{j=1}^{C}\left(X_{k,j} \times D_{k,j,t} + F_k\right) = D_{k,t}$$

$$X_{k,j} \geq 1$$

The capture ratios $CR_{k,j}$ are given by:

$$CR_{k,j} = \frac{\sum_{i=1}^{C}\sum_{t=1}^{T} D_{k,i,t}}{X_{k,j} \times \left(\left[\sum_{i=1}^{C}\sum_{t=1}^{T} D_{k,i,t}\right] + F_k\right)}$$

This solution reduces to the one without a noise factor in the limit $F_k \to 0$. Finally, the presence of workloads with small measurement data also influences negatively this type of capture ratio determination. It is recommended to group multiple small workload types into one single workload type.

The main disappointing factor about this approach may be the black box character of it : the proposed method does not give a real insight in the nature of the capture ratios and the differences in capture ratios amongst the workload types. To solve this, an activity based costing approach is developed next.

Activity Based Costing techniques for capture ratio determination

In cost accounting, Activity Based Costing assigns capacity utilization to activities not on the basis of measured utilization, but by means of significant activity factors that can explain the capacity utilization [Horngren, Foster & Datar 94, Cooper & Kaplan 92]. The application of the same ideas to capture ratio determination is very natural. The activity that causes capture ratios in performance measurements is operating system activity in supervisor mode, such as I/O, Paging and Swapping. Suppose that these types of activities can be measured in terms of system events, grouped by type. In concrete, let

$E_{k,j,l}$ = the number of system events of type l involving device k as measured for some workload type j

The basic idea behind this lies in the fact that "counting" system events may be easier than measuring service demands, so that the system event counting is supposed to be more correct (meaning, closer to reality) than service demand measurement data. Suppose furthermore that each event type l involving device k receives a weighting factor $W_{k,l}$. The procedure to determine the capture ratios by means of system events proceeds as follows. First, define

$$E_{k,j} = \sum_{l=1}^{L} \left(E_{k,j,l} \times W_{k,l} \right)$$

where L is the total number of system event types, and

$$E_k = \sum_{j=1}^{C} E_{k,j}$$

With the same notations as used in the global capture ratio determination method, the corrective portion per service demand for device k by workload type j is given by :

$$Dovh_{k,j} = \left[\frac{E_{k,j}}{E_k} \right] \times D_{k,ovh}$$

Hence, the capture ratios are given

$$CR_{k,j} = \frac{D_{k,j}}{D_{k,j} + \left(\frac{E_{k,j}}{E_k} \right) \times D_{k,ovh}}$$

A major advantage of this approach is the possibility that it creates to distribute the non-captured service demands by event type. The portion of the non-captured service demand on device k due to system events of type l is given by

$$D_{k,l,ovh} = D_{k,ovh} \times \left(\frac{\sum_{j=1}^{C} E_{k,j,l} \times W_{k,l}}{E_k} \right)$$

The weighting factors $W_{k,l}$ play a crucial role in this approach. They can be determined by constructing benchmarks with dedicated workloads. Alternatively, the weighting factors can also be obtained from the solutions of the following multiple regression model :

$$\sum_{l=1}^{L} \left[Y_{k,l} \times E_{k,l,t} + Z_k \right] = D_{k,ovh,t}$$

$$Y_{k,l} \geq 0$$

where t indicates a time interval during which the elements $D_{k,ovh}$ and $E_{k,l}$ have been measured. To have a meaningful multiple regression model, at least L time intervals with measurements are needed. Z_k is again a noise factor, which increases the independency amongst the unknowns in the regression model, as discussed before. When a solution of the above problem is obtained, the capture ratio determination is done as follows :

$$D_{k,l,ovh} = Y_{k,l} \times \left(\sum_{j=1}^{C} E_{k,j,l} \right) + Z_k \times \left(\frac{Y_{k,l}}{\sum_{l=1}^{L} Y_{k,l}} \right)$$

$$W_{k,l} = \frac{D_{k,l,ovh} \times E_k}{D_{k,ovh} \times \left(\sum_{j=1}^{C} E_{k,j,l} \right)}$$

Of course, the remarks formulated on the usage of multiple regression models under boundary conditions in the previous section are still valid. Observe as a final remark that only the "relative" weighting factors are important : multiplying all weighting factors by the same factor does not change the results.

A case study on CPU capture ratios

The different methods that have been described so far are now illustrated with a case study on capture ratio determination for non-captured CPU service demands in a mainframe server configuration. The following workload types have been considered :

Time Sharing : a closed terminal workload type, where the time sharing transactions are using a priority feedback mechanism. This means that short time sharing transactions are executed with a period 1, while transactions with more service demands migrate to periods 2 and 3, with lower service levels. Period 3 performance is almost like short Batch jobs. In this case study, the workload is a data entry workload that has short think times and a high number of terminals.

Batch : A closed Batch type of workload.

Database-Online : An open workload representing a database management system, that is used in an interactive operation.

Database-Batch : A closed Batch workload representing a database management system, that is used in a batch mode operation.

Other : This workload gathers all the other activity on the servers (such as system tasks).

Two mainframe servers with a different mix of these workloads have been considered. The first server is more database intensive, while the second one has an emphasis on Time Sharing and Batch. The following table gives an overview of the relative importance of the workloads :

% of CPU utili-zation	**Time-Sha-ring Period 1**	**Time-Sha-ring Period 2**	**Time-Sha-ring Period 3**	**Data-base-Batch**	**Data-base-On-line**	**Batch**	**Other**
Server 1	1.02	1.10	1.74	11.69	53.44	7.32	23.69
Server 2	5.03	6.75	23.46	2.83	6.78	32.23	22.92

The methods from the previous sections are now specialized into five variations for this case, as follows.

METHOD A :

This method leads to the same capture ratios for all workloads, namely

Server 1 : $CR_{CPU,tot} = 0.82$
Server 2 : $CR_{CPU,tot} = 0.81$

Capture ratio calculations for mainframe servers according to this method have been proposed in [Wicks 89, Irwin 83]. Critical remarks have been formulated when discussing global capture ratio calculations. The fact that capture ratios from this method don't give an insight is also illustrated by the fact that the two global capture ratios are almost the same on both servers, even with a significantly different workload mix.

METHOD B :

This method is based on multiple regression. Although some suggestions to use this type of approach for mainframe servers was published in [Wicks 89], the problems related to the boundary conditions and the noise factor solutions were not presented. In this case, the following model was solved for each server :

$$\sum_{j=1}^{C}\left(X_j \times CPUtime_{j,t} + F\right) = TotalCPU_t$$
$$X_j \geq 1$$

for a sufficient number of time intervals t. The results of this method are the following :

CPU capture ratio	**Time-Sharing Period 1**	**Time-Sharing Period 2**	**Time-Sharing Period 3**	**Data-base-Batch**	**Data-base-On-line**	**Batch**	**Other**
Server 1	0.5667	0.7170	0.7845	0.8429	0.8521	0.8412	0.7130
Server 2	<u>0.6791</u>	<u>0.6714</u>	0.9168	0.8459	0.8149	0.8808	0.7546

There is an "anomaly" in the results for server 2. As a time sharing transaction migrates from period 1 to period 3 over period 2, the capture ratio should increase, as long running transactions perform almost as Batch jobs. On server 2 the capture ratio for period 2 is not satisfying this. The case study also revealed that it was hard to get stable results from the regression exercises: a small change in the measurement data resulted sometimes in large fluctuations in the capture ratios.

METHOD C :

The idea of using system events for capture ratio determination was suggested in [Kuchnicki 81, Artis 89 and Lazowska 84], but has never been elaborated afterwards. Contemporary mainframe server monitors allow to include the following event types in capture ratio determination:

1 : Physical Swapping to expanded memory
2 : Physical Swapping to Page/Swap files on disk storage
3 : Logical Swapping (working set is kept in main memory)
4 : Swap Paging to expanded memory
5 : Swap Paging to Page/Swap files on disk storage
6 : Non-Swap Paging to expanded memory
7 : Non-Swap Paging to Page/Swap files on disk storage
8 : Input/Output to disk storage

In method C, the weighting factors have been derived from published benchmarks on dedicated workloads [Kuchnicki 81, Bank 90]. The weighting factors are :

$W_{CPU,1} = 11.35$; $W_{CPU,2} = 45.4$; $W_{CPU,3} = 18.6$; $W_{CPU,4} = 0.34$;
$W_{CPU,5} = 1.36$; $W_{CPU,6} = 0.67$; $W_{CPU,7} = 2.7$; $W_{CPU,8} = 1.6$

The results for this method are

CPU capture ratio	**Time-Sharing Period 1**	**Time-Sharing Period 2**	**Time-Sharing Period 3**	**Data-base-Batch**	**Data-base-On-line**	**Batch**	**Other**
Server 1	0.6043	0.6129	0.6527	0.7713	0.8950	0.6795	0.7780
Server 2	0.4334	0.7325	0.8964	0.9107	0.8710	0.8768	0.8883

The results are stable, but show some very low capture ratios for the first period of the time share interactive workload. There are no anomalies and Time Sharing period 3 capture ratios are very close to the Batch capture ratios.

METHOD D :

This method is completely the same as method C, but a multiple regression model with boundary conditions has been used to determine the weighting factors. The capture ratio results are the following :

CPU capture ratio	**Time-Sharing Period 1**	**Time-Sharing Period 2**	**Time-Sharing Period 3**	**Data-base-Batch**	**Data-base-On-line**	**Batch**	**Other**
Server 1	0.4486	0.6064	0.7330	0.7376	0.9022	0.6761	0.8385
Server 2	0.5709	0.7547	0.8789	0.9006	0.8386	0.8468	0.8465

The results of this method are also stable, and are close to those from method C for larger workload types. There are no anomalies across the workload capture ratios.

METHOD E :

This method only considers three system event types and was suggested in [Artis 89 and Lazowska 84]. The event types

1 : Physical Swapping
2 : Total Paging
3 : Input/Output to disk storage

In analogy with method D, the weighting factors have been derived from a multiple regression problem. An analogous problem was formulated in [Artis 89], however without boundary conditions. The results for this method are

CPU capture ratio	**Time-Sharing Period 1**	**Time-Sharing Period 2**	**Time-Sharing Period 3**	**Data-base-Batch**	**Data-base-On-line**	**Batch**	**Other**
Server 1	<u>0.6562</u>	<u>0.6132</u>	0.6679	0.7971	0.8783	0.6443	0.7852
Server 2	0.7150	0.7567	0.8795	0.8756	0.7997	0.7978	0.8722

This method is suspicious as it does not take into consideration the subtle differences between using expanded memory and disk storage for Page and Swap activity. Moreover, the results suffer again from anomalies.

Discussion

The research that resulted in this paper was mainly motivated by the search for more stable capture ratios for service demands in multiple workload performance measurements. The case study demonstrated how Activity Based Costing techniques can succesfully be applied to this problem. The activities that are considered are the relevant system events that give rise to non-captured service demands. The case study results, in particular for method E, demonstrated that the choice of system events cannot be oversimplified. The case study also revealed that for some workload types, such as interactive workloads, capture ratios can be very significant. As a result, correct performance modeling without capture ratios seems impossible.

The Activity Based Costing techniques allow furthermore to assign the non-captured service demands to particular system events. This provides additional instruments to performance analysts for predicting the results from changes in the tuning of a configuration.

Although the case study in this paper concentrated on CPU measurements, the results can be extended to memory and disk service demands. Ideally, capture ratio determination should also be included in monitors.

References

[Artis 89] P. Artis, "Capture Ratios : Fact or Legend", Mainframe Journal 1989.

[Bank 90] J.H. Bank, "SMF Accounting for data in memory", SHARE 75 Proceedings, 1990.

[Bourret 80] P. Bourret & Cros P., "Presentation and Correction of Errors in Operating Systems Measurements", IEEE Transactions on Software Engineering, 1980.

[Cooper & Kaplan 92] R. Cooper & R.S. Kaplan, "Activity-Based Systems : Measuring the Costs of Resource Usage", Accounting Horizons, 1992.

[Hongren, Foster and Datar 94] C. T. Hongren, G. Foster and S. Datar, "Cost Accounting : a Managerial Emphasis", Prentice-Hall, Englewood Cliffs, 1994.

[Irwin 83] R. Irwin, "RMF Equations : obtaining job class level results from RMF", Journal of Capacity Management, 1983.

[Kuchnicki 81] P. Kuchnicki, "CPU overhead analysis", Federal Reserve Bank of St. Louis Working Paper, CMG Proceedings, 1981.

[Lazowska 84] E. Lazowska, J. Zahorjan, G. Graham & K. Sevcik, "Quantitative System Performance", Prentice Hall 1984.

[Wicks 89] R. Wicks, "Capacity planning with RMF Data", SHARE 72 Proceedings, 1989.